

AMENDMENTS IN THE CLAIMS

1. (currently amended) In a computer system having I/O components and a file system existing within a volume group comprised of storage media, a method for substantially preventing I/O failure due to insufficient storage space within the file system, said method comprising:

determining that a received I/O operation directed at said file system requires more storage space than is currently available within said file system;

dynamically expanding the storage space available within said file system to increase the size of the file system based on the amount of additional space required within the file system to accommodate said I/O operation, wherein additional space on said volume group is allocated to said file system only when the size of an existing space on the file system is not sufficient to accommodate the I/O operation; and

subsequently completing said I/O operation within said file system;

wherein said dynamically expanding step and said subsequently completing step are both completed without user input and/or activation.

2. (original) The method of Claim 1, wherein said dynamically expanding step includes assigning reserve storage space existing within said volume group to a logical volume hosting said file system.

3. (original) The method of Claim 1, wherein said subsequently completing step comprises restarting said I/O operation within kernel space without requiring user input.

4. (original) The method of Claim 1, further comprising issuing a notification indicating that said dynamically expanding step is being completed.

5. (currently amended) The method of Claim 1, further comprising:

signaling a logical volume manager (LVM) of a need for a specific size of additional storage space within the file system for completing said I/O operation; and

completing an automatic expansion of a logical volume hosting said file system, wherein said dynamically expanding step expands said file system into at least the specific size of available space within said logical volume following said automatic expansion.

6. (currently amended) The method of Claim 5, wherein said signaling step is completed via an I/O failure response (FR) daemon that coordinates communication between control blocks in [[the]] a kernel space and the LVM.

7. (currently amended) The method of Claim 1, wherein said determining step comprises:
parsing parameters from [[said]] an I/O command for a size of [[said]] required storage space required to complete said I/O operation; [[and]]
comparing said required storage space with an available storage space size within said file system; and
triggering the dynamic expansion only when the size of available storage space is less than the required storage space.

8. (currently amended) The method of Claim 1, wherein said dynamically expanding step comprises:
determining that said additional storage space is available within [[said]] a reserve space;
and
expanding said file system to include a preset amount of space from said reserve space.

9. (currently amended) The method of Claim [[5]] 8, wherein said expanding step includes iteratively expanding said file system by said preset amount of space until a total space within said file system is sufficient to accommodate said I/O operation.

10. (currently amended) The method of Claim 1, wherein said dynamically expanding step comprises:
calculating an amount of additional space required to complete said I/O operation, with consideration of currently available space within said file system ; and
dynamically expanding said file system by at least said amount of additional space required.

AUS920030447US1

Amendment A

10/666,795

-3-

11. (currently amended) In a computer system having I/O components and a file system existing within a volume group comprised of storage media, a system for mitigating I/O failure due to insufficient storage space within the file system, said system comprising:

means for determining that a received I/O operation directed at said file system requires more storage space than is currently available within said file system;

means for dynamically expanding the storage space available within said file system to increase the size of the file system based on the amount of additional space required within the file system to accommodate said I/O operation, wherein additional space on said volume group is allocated to said file system only when the size of an existing space on the file system is not sufficient to accommodate the I/O operation; and

means for subsequently completing said I/O operation within said file system;

wherein said means for dynamically expanding and said means for subsequently completing both initiate without user input and/or activation.

12. (original) The system of Claim 11, wherein:

said means for dynamically expanding includes means for assigning reserve storage space existing within said volume group to a logical volume hosting said file system; and

said means for subsequently completing comprises means for restarting said I/O operation within kernel space without requiring user input.

13. (original) The system of Claim 11, further comprising means for issuing a notification indicating that said dynamically expanding step is being completed.

14. (currently amended) The system of Claim 11, further comprising:

means for signaling a logical volume manager (LVM) of a need for a specific size of additional storage space within the file system for completing said I/O operation; and

means for enabling said LVM to complete an automatic expansion of a logical volume hosting the file system, wherein said LVM signals said file system of a completion of said automatic expansion, and wherein said dynamically expanding means expands said file system into at least the specific size of available space within said logical volume following said automatic expansion.

AUS920030447US1

Amendment A

10/666,795

-4-

15. (currently amended) The system of Claim 11, further comprising an I/O failure response (FR) daemon that coordinates communication between control blocks in [[the]] a kernel space and [[the]] a logical volume manager (LVM).

16. (currently amended) The system of Claim 11, wherein said means for dynamically expanding comprises:

means for determining that said additional storage space is available within [[said]] a reserve space; and

means for expanding said file system to include a preset amount of space from said reserve space, wherein said means for expanding reiteratively expands said file system by said preset amount of space until a total space within said file system is sufficient to accommodate said I/O operation.

17. (currently amended) The system of Claim 11, wherein said dynamically expanding [[step]] means comprises:

means for calculating an amount of space required to complete said I/O operation given a value of currently available space within said file system ; and

means for dynamically expanding said file system by at least said amount of space required.

18. (currently amended) The system of Claim 11, wherein:

said means for determining includes an I/O controller [[CC]] and operating system (OS) functional logic, said means further comprising:

means for parsing parameters from an I/O command for a size of required storage space to complete said I/O operation;

means for comparing said required storage space with an available storage space size within said file system; and

means for triggering the dynamic expansion only when the size of available storage space is less than the required storage space;

said means for dynamically expanding includes [[said]] a logical volume manager (LVM); and

AUS920030447US1

Amendment A

10/666,795

-5-

said means for notifying includes an I/O failure response (FR) daemon that bridges communication between said I/O [[CC]] controller at an OS level and said LVM at an application level within said computer system.

19. (currently amended) A computer program product comprising:

a tangible computer readable medium; and

computer program code on said tangible computer readable medium for substantially preventing I/O failure due to storage space restrictions within a file system, said program code further comprising code for:

determining that a received I/O operation directed at said file system requires more storage space than is currently available within said file system;

dynamically expanding the storage space available within said file system to increase the size of the file system based on the amount of additional space required within the file system to accommodate said I/O operation, wherein additional space on said volume group is allocated to said file system only when the size of an existing space on the file system is not sufficient to accommodate the I/O operation; and

subsequently completing said I/O operation within said file system.

20. (currently amended) The computer program product of Claim 19, further comprising code for:

assigning reserve storage space existing within [[said]] a volume group to a logical volume hosting said file system; and

restarting said I/O operation within kernel space without requiring user input;

wherein said code for implementing said dynamically expanding step and said subsequently completing step are executed without user input and/or activation. [[.]]

21. (currently amended) The computer program product of Claim 19, further comprising code for:

implementing an I/O failure response (FR) daemon that coordinates communication between control blocks in [[the]] a kernel space and [[the]] a logical volume manager (LVM), wherein said I/O FR completes a set of functional operations including:

signaling a logical volume manager (LVM) of a need for additional storage space for completing said I/O operation;

issuing a notification indicating that said dynamically expanding step is being completed; and

initiating a restart of said I/O operation once said expansion completes;

wherein said LVM completes an automatic expansion of a logical volume hosting said file system, and said dynamically expanding step expands said file system into available space within said logical volume following said automatic expansion.

22. (currently amended) The computer program product of Claim 19, wherein said code for determining comprises additional code for:

parsing parameters from [[said]] an I/O command for a size of required storage space ~~required~~ to complete said I/O operation; [[and]]

comparing said required storage space with an available storage space size within said file system; and

triggering the dynamic expansion only when the size of available storage space is less than the required storage space.

23. (currently amended) The computer program product of Claim 19, wherein said code for dynamically expanding comprises code for:

determining that said additional storage space is available within [[said]] a reserve space; expanding said file system to include a preset amount of space from said reserve space; and

iteratively expanding said file system by said preset amount of space until a total space within said file system is sufficient to accommodate said I/O operation.

24. (currently amended) The computer program product of Claim 19, wherein when there is not sufficient space within [[said]] a reserve space, said code comprises additional code for signaling a complete failure of said I/O operation.

25. (currently amended) The computer program product of Claim [[16]] 19, wherein said code for dynamically expanding step comprises:

AUS920030447US1

Amendment A

10/666,795

-7-

code for calculating an amount of space required to complete said I/O operation given a value of currently available space within said file system ; and

code for dynamically expanding said file system by at least said amount of space required.